**Имена:**  Адриян Ибовски

**фн:**  *82157***Начална година:** *2020* **Програма:** бакалавър, (КН) **Курс: 4**   
**Тема: Filemeup  
Дата: 2024-02-08 Предмет: w24prj\_KN \_final имейл: adriyan.ibovski@gmail.com**

**Имена:**  Емил Йорданов

**фн:**  *82193***Начална година:** *2020* **Програма:** бакалавър, (КН) **Курс: 4**   
**Тема: Filemeup  
Дата: 2024-02-08 Предмет: w24prj\_KN \_final имейл: zyppyvids@gmail.com**

**Имена:**  Ралица Трендафилова

**фн:**  *82152***Начална година:** *2020* **Програма:** бакалавър, (КН) **Курс: 4**   
**Тема: Filemeup  
Дата: 2024-02-08 Предмет: w24prj\_KN \_final имейл: rally0307@gmail.com**

Filemeup Project Documentation

# ТЕМА: w24/7 Filemeup

# 1. Условие

# Filemeup е уеб базирано приложение, предназначено за качване, съхраняване и преглед на файлове, подобно на услуги като Google Drive. Той позволява на потребителите да качват файлове на сървър, да ги управляват и преглеждат онлайн.

# 2. Въведение – извличане на изисквания

# Идентифицираните роли включват регистрирани потребители. Функционалните изисквания включват управление на файлове, удостоверяване на потребителя и сигурност на данните. Нефункционалните изисквания включват отзивчивост, мащабируемост и използваемост.

# 3. Теория – анализ и проектиране на решението

Приложението Filemeup е структурирано в няколко основни модула, които заедно осигуряват пълна функционалност за удостоверяване на потребителя, управление на файлове и преглед на файлове. Всяка част от приложението е внимателно проектирана, за да отговаря на нуждите на потребителите и да осигури сигурност и удобство при употреба.

* **Модул за Удостоверяване на Потребителя**

Този модул е отговорен за всички аспекти на удостоверяването на потребителя, включително регистрация, вход в системата и управление на потребителски профили. Той осигурява сигурност чрез силни мерки за аутентикация и защита на личните данни. Също така включва функционалности за възстановяване на забравени пароли и защита срещу неоторизиран достъп.

* **Модул за Управление на Файлове**

Този модул позволява на потребителите да качват, съхраняват, организират и изтриват файлове. Включва функции за категоризиране на файлове по тип, размер и дата на създаване. Също така предлага опции за споделяне на файлове с други потребители и настройки за достъпност.

* **Модул за Преглед на Файлове**
  + **Подмодул за частни (“private”) файлове**

Този модул дава възможност за преглед на качените частни файлове директно в браузъра. Той поддържа различни формати на файлове и предоставя опции за преглед на изображения, документи и медийни файлове. Включва също функционалности за изтриване, изтегляне и преглед на файлове в рамките на потребителския интерфейс.

* + **Подмодул за публични (“public”) файлове**

Този модул дава възможност за преглед на споделените файлове директно в браузъра. Той поддържа различни формати на файлове и предоставя опции за преглед на изображения, документи и медийни файлове. Включва също функционалности за изтегляне и преглед на файлове в рамките на потребителския интерфейс.

**Интеграция и Съвместимост**

В допълнение към гореспоменатите модули, системата е проектирана да бъде лесно интегрируема с други системи и платформи. Това включва възможност за взаимодействие с различни бази данни и поддръжка на най-разпространените уеб браузъри. Също така, дизайнът на системата е отзивчив и адаптивен, което я прави подходяща за използване на различни устройства - от настолни компютри до мобилни телефони и таблети.

# 4. Използвани технологии

# Използваните технологии включват HTML, CSS, JavaScript за разработка на front-end-a, PHP за скриптове от страна на сървъра и SQL (MariaDB) за управление на бази данни, за страна на сървъра XAMPP - Apache. Приложението е съвместимо с основните уеб браузъри и изисква уеб сървър за внедряване.

# 5. Инсталация, настройки и DevOps

# Проектът съдържа следната файлова структура:

# [Проект]

# [css] [js] [php] [img]

# Освен това има 4 html файла - fileviewer.html, login.html, main.html и register.html. index.php отговаря за началното пренасочване след влизане в основната страница.

# Папка css съдържа всички стилизиращи файлове.

# Папка js съдържа всички javascript файлове.

* Папка php съдържа всички php файлове
* Папка img съдържа изображения, използвани в изгледите на приложението.

Възможна е промяна към докер контейнери, тъй като те предлагат по-високо ниво на гъвкавост.

За инсталиране на XAMPP - [https://www.apachefriends.org/download.html](https://www.apachefriends.org/download.html?fbclid=IwAR2GEZt7YpKgNZS7ZHXbMjmspFisCx2MpnRLVmjx8Gxrp6oAQLMNF9Eg1IA)

**6. Кратко ръководство на потребителя**

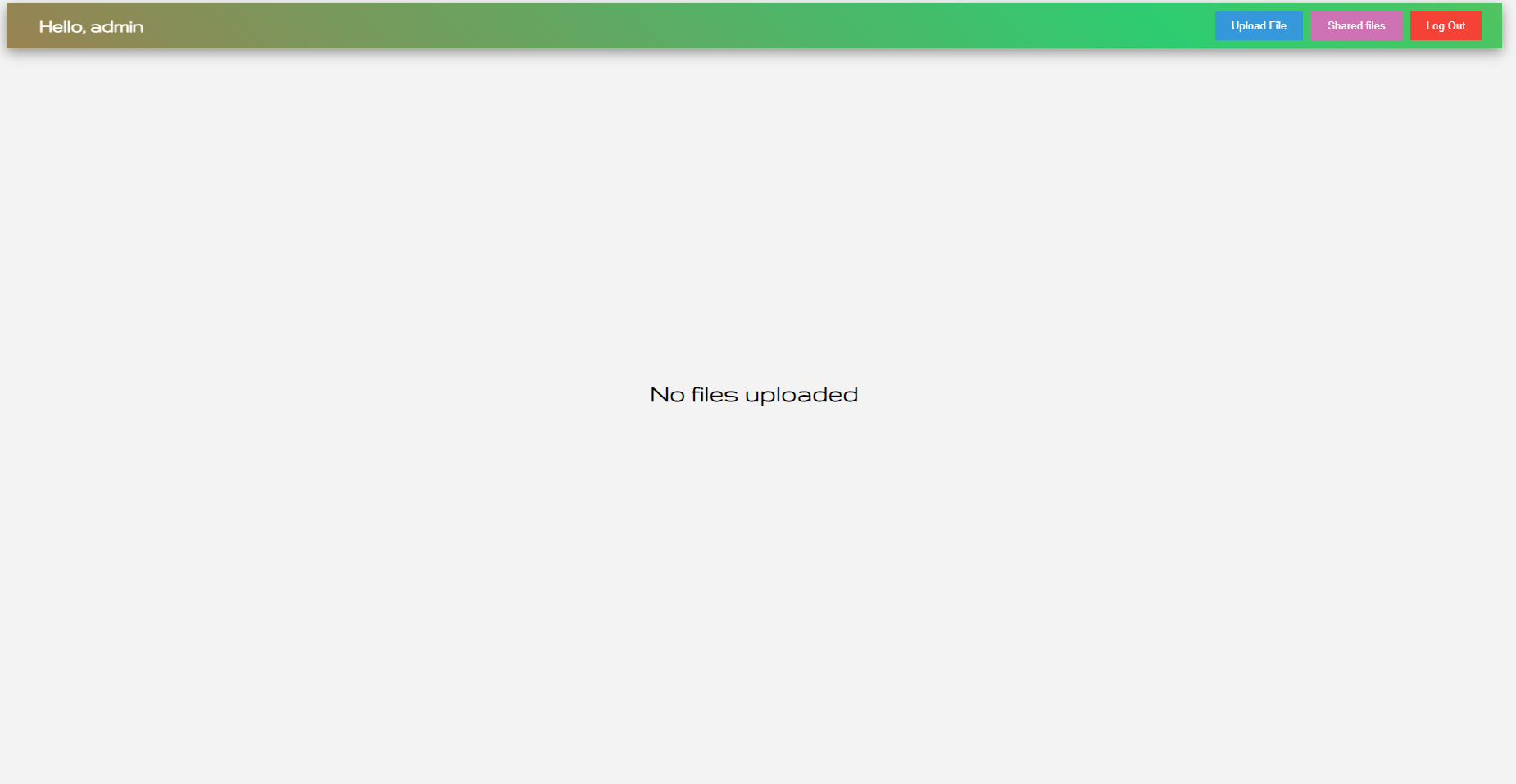
След успешно инсталиране на приложението, първата страница, с която ще се запознае потребителя е login/ register.



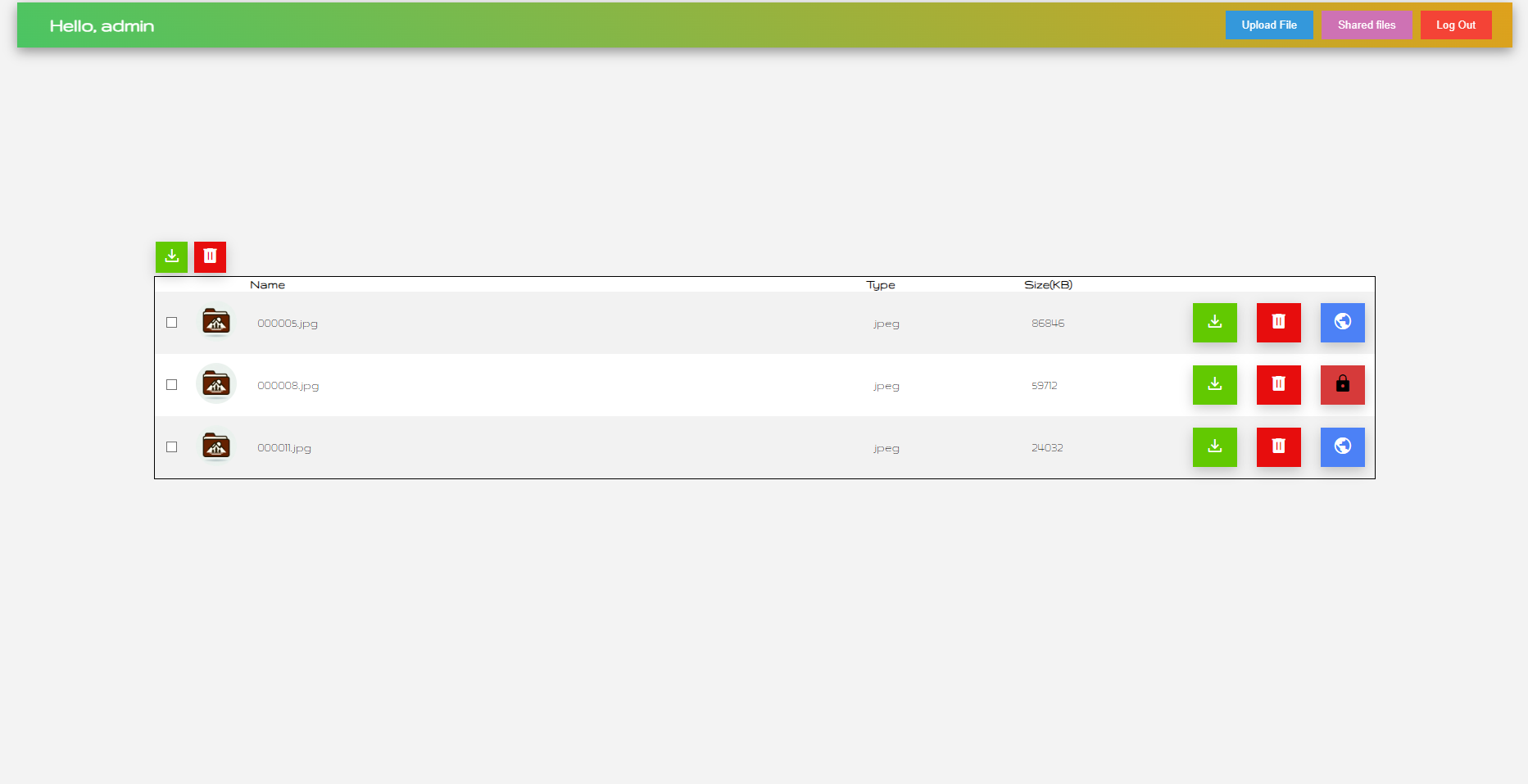


Потребителят трябва да въведе username и password полетата, като паролата задължително трябва да съдържа главни, малки букви и цифри. Също така да е поне 8 символа.

След успешен логин потребителя ще срещне главаната страница, където е и основната функционалност на FilemeUp приложението.

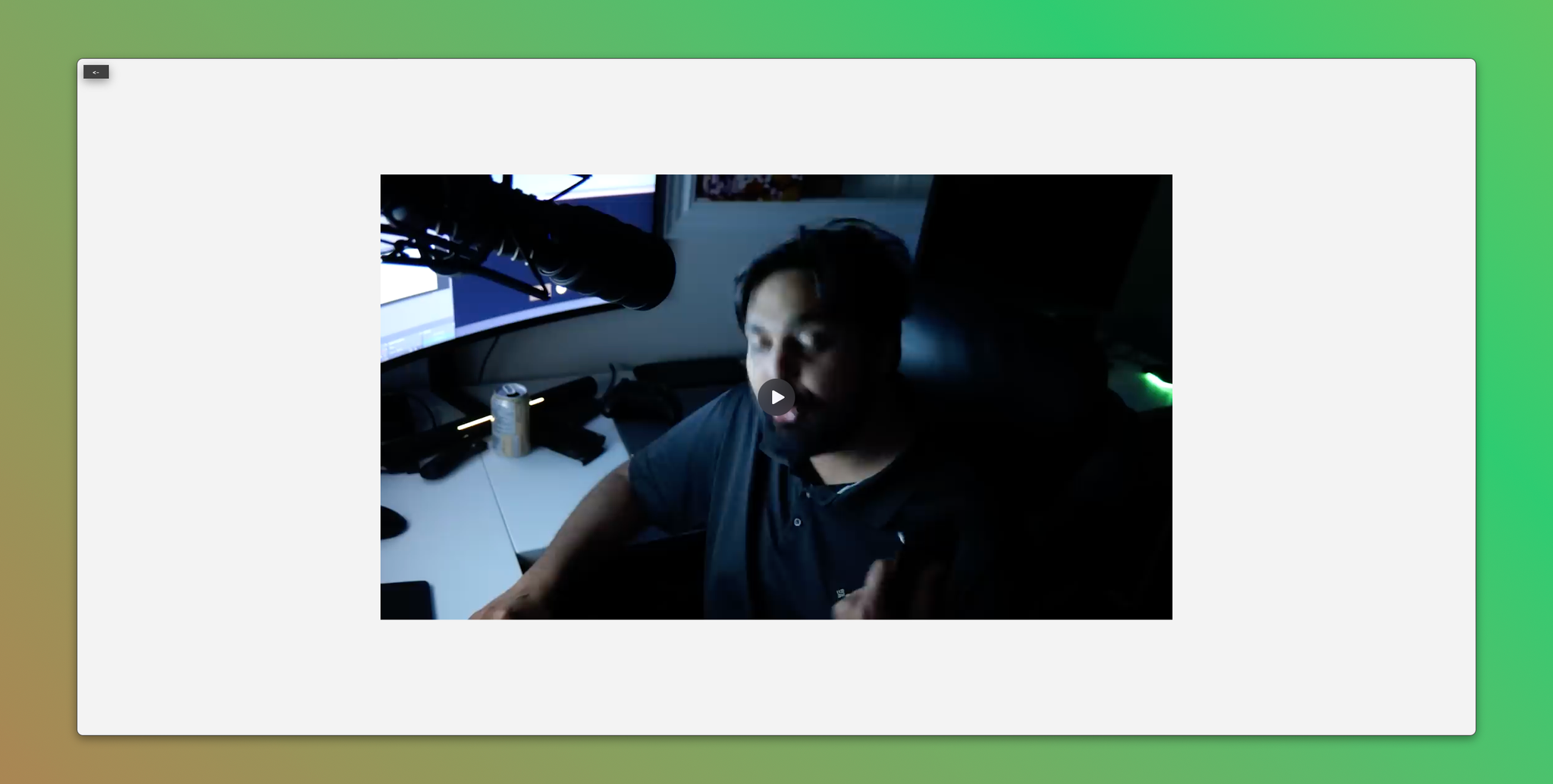


Потребителя започва с празен лист от файлове, тъй като още нищо не е качил в системата. С бутона “Upload” може да си избере файл или множество от файлове и да ги качи в сървъра.

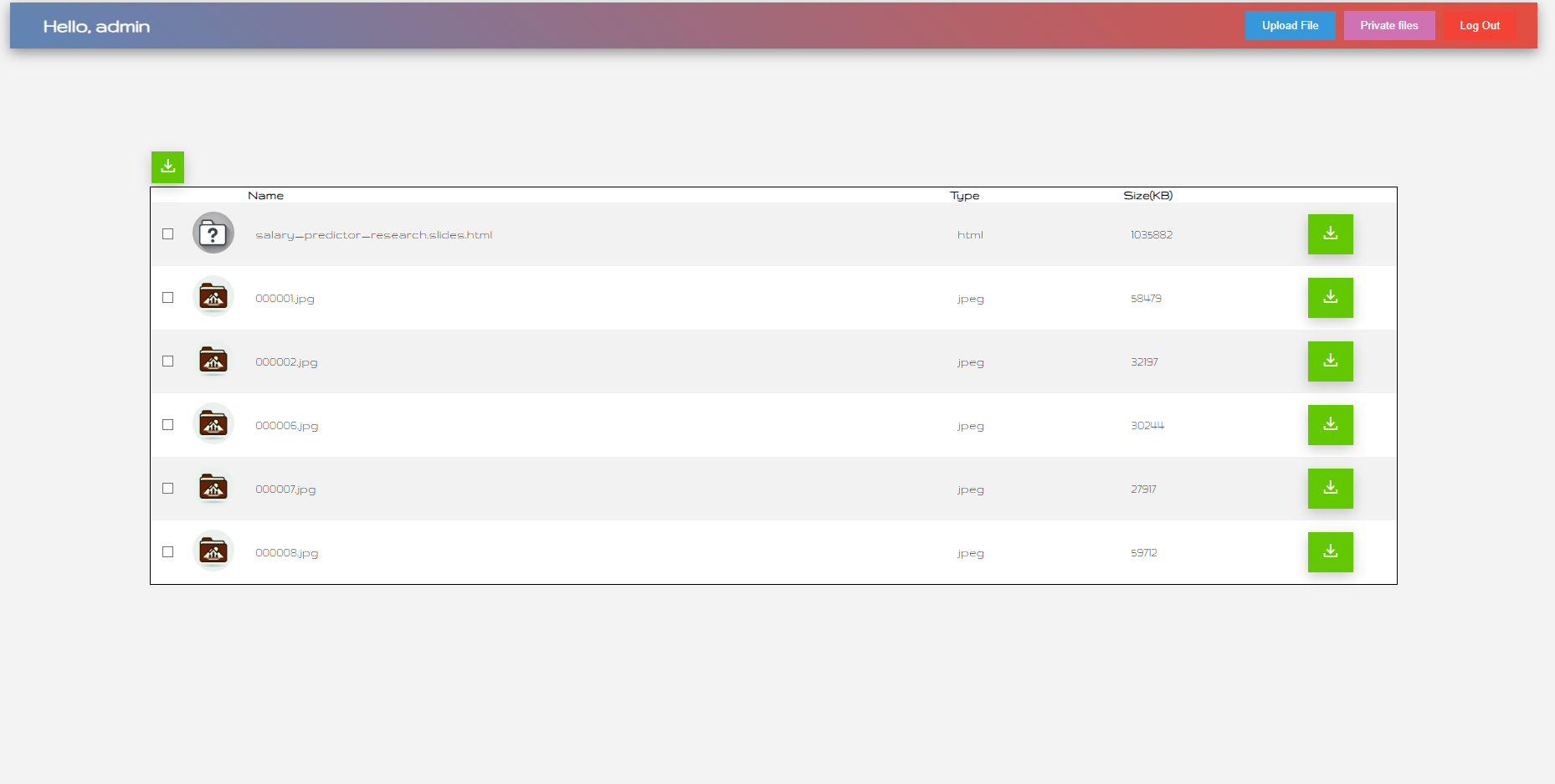


Системата поддържа огромен брой типове файлове като jpeg, png, pdf, txt, mp4, video и други като файла не трябва да надвишава 5MB.

След успешното качване на файл в системата, потребителя може по всяко време да прегледа този файл, да го изтегли или изтрие от системата. Отварянето на файл става с кликането върху името му.



Също така ако желае, потребителят може да направи някой от файловете си публични, достъпни за всички останали юзери. Споделените файлове може да се видят в специален изглед „Shared Files”.



# 7. Входни данни

За тестване на системата се изискват следните неща:

* Username – уникален.
* Password (Задължително поне 8 символа, в които една главна, малка буква и цифра).
* Произволен файл, чиито размер е не по – голям от 5MB.

# 8. Описание на програмния код

Програмният код се състои от HTML, CSS, JavaScript и PHP файлове.

HTML файловете са 4 - един за login екрана, един за register екрана, един за основна страница (страница с всички качени файлове на конкретния потребител) и един за преглеждане на даден файл.

CSS файловете са 4 – login и register споделят един общ, един за основната страница, един за страницата с изглед към файл и един за „snackbar”, който използваме за изобразяване на грешки в ситемата.

PHP файловете са 9:

* index.php - действа като основна страница при влизането на основната страница на сървъра;
* connection.php - съдържа логиката и настройките нужни за създаване на връзка с БД;
* login.php - съдържа основната логика за проверка на данните на потребителя при влизане в профила му;
* register.php - съдържа основната логика за проверка и създаване на нов профил на потребител;
* upload.php - съдържа основната логика за качване на файл както на сървъра (в папка ./uploads), така и в БД като ред в таблицата files;
* fetch\_data.php - съдържа основната логика за вземане на данните от таблица files за конкретен потребител.;
* delete.php – съдържа основната логика за премахване на файл както от сървъра така и от БД;
* get\_file\_visibility.php – съдържа основната логика за проверка на видимостта на файл (публичен или частен);
* update\_visibility.php – съдържа основната логика за обновяване на видимостта на файл (публичен или частен).

JavaScript файловете са 13:

* login.js - съдържа клиент логиката за извикването на login.php;
* register.js - съдържа клиент логиката за извикването на register.php;
* logout.js - съдържа клиент логиката за излизане от конкретния профил (в случая използваме локалната сесия като начин на оторизация, но това винаги може да се промени на cookie-based оторизация **например**);
* upload.js - съдържа клиент логиката за извикването на upload.php;
* filefunc.js - съдържа клиент логиката при избирането на файл за изобразяване (логиката за отваряне на файл, за селектиране на файлове и действията свързани със селектирани файлове);
* fileviewer.js - съдържа клиент логиката за правилно изобразяване на конкретен файл в fileviewier.html;
* fetchdata.js - съдържа клиент логиката за извикването на fetch\_data.php и изобразяването на резултата (файловете на конкретния потребител);
* backbutton.js - съдържа клиент логиката на бутона “назад”, намиращ се в fileviewer.html;
* preview.js – съдържа клиент логиката за “preview” на изображение (ако има такова) във файловете на потребителя;
* snackbar.js – съдържа клиент логиката за изобразяване на потребителски и сървърни грешки (като поле в дъното на страницата);
* toggle.js – съдържа клиент логиката за сменяне на изгледа на таблицата с файлове (частни или публични);
* validation.js – съдържа клиент логиката за валидиране на парола на потребител (при вход / регистрация);
* windowfunc.js – съдържа клиент логиката за опресняване на страница.

# 9. Приноси на студента, ограничения и възможности за бъдещо разширение

# Емил Йорданов - Обща функционалност и функционалност на клиентската част от проекта;

# Адриян Ибовски - Обща функционалност и стилизация;

# Ралица Трендафилова- Login/Register функционалност;

Бъдещо развитие - Използване на облачно пространство за съхранение на файловете (например AWS), подобрения в сигурността.

# 10. Какво научих?

**Студент 1: Адриян Ибовски**

През целия курс и разработването на проекта Filemeup, Адриян придоби значителни умения в уеб разработката, с фокус върху основите на HTML и CSS за структуриране и стилизиране на уеб страници. Той разви добро разбиране за позициониране на елементи и основните принципи на CSS.. Освен това, Адриян разви своите умения в PHP и JavaScript, което му позволи да разбере по-добре клиент-сървър взаимодействията и асинхронното програмиране. Работата върху проекта също така му даде възможност да се запознае с управлението на бази данни с MySQL, което беше ново и вълнуващо предизвикателство.

**Студент 2: Ралица Трендафилова**

Ралица беше отговорна за сигурността и аутентикацията на потребителите в приложението. Тя научи важни аспекти на киберсигурността, като хеширане на пароли и защита от SQL инжекции. Работата върху проекта и курса я запозна с различни методи за сигурно управление на потребителски сесии и защита на чувствителна информация. Ралица също така разви своите умения в PHP, особено в контекста на създаване на сигурни форми за вход и регистрация.

**Студент 3: Емил Йорданов**

Емил фокусираше върху клиентската страна на проекта, като особено внимание отделяше на JavaScript и фронтенд логиката. Той научи как да използва асинхронно зареждане на данни с AJAX, което подобри потребителското изживяване и ефективността на приложението. Емил също разшири своите знания за DOM манипулации и усъвършенства уменията си в откриването и отстраняването на грешки в JavaScript кода. Той също така научи значението на добре структурирания и чист код за поддръжката и разширяемостта на уеб приложения.

# 11. Използвани ресурси

**PHP**:

* PHP Documentation: [https://www.php.net/manual/en/](https://www.php.net/manual/en/?fbclid=IwAR2jWCAQjFiwdZGomsjklqyzYAWJVVsxjw5VKE5Mfh0IPyDY6E8BpGWE8Uc)
* PHP File Handling: [https://www.w3schools.com/php/php\_file.asp](https://www.w3schools.com/php/php_file.asp?fbclid=IwAR1zCgE_P1enq0PuI-oDDogbzR8_KOiQQExm19FOk7gTa0FXipfsveVszWk)

**HTML/CSS**:

* HTML and CSS Tutorials: [https://www.w3schools.com/html/](https://www.w3schools.com/html/?fbclid=IwAR2_wh5lE5yOZvIM2zCeMVrIqoRZ9Np9zfowbZhIbtatJb4rD9D8nd5V0Tk)

**JavaScript**:

* JavaScript MDN Web Docs: [https://developer.mozilla.org/en-US/docs/Web/JavaScript](https://l.facebook.com/l.php?u=https%3A%2F%2Fdeveloper.mozilla.org%2Fen-US%2Fdocs%2FWeb%2FJavaScript%3Ffbclid%3DIwAR1UT1DLbG5nJW54CuW3hjXKHmEw5UO2a9At7YhK3K90fr8uHV7EIeYOCDA&h=AT3IMxL9uAtSwd0dcfrJtmZmA5DtlQUDgegZc0T0T0XYxpkILM5CnIVIIKNR_g0PYhQ0udGqbOukfs1OLjvkH6vzzEPwTo2Bi1yZPgA1dnZe1pMGlcZZjpHpv6frbmfia-RCAg)
* JavaScript and the DOM: [https://www.w3schools.com/js/js\_htmldom.asp](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.w3schools.com%2Fjs%2Fjs_htmldom.asp%3Ffbclid%3DIwAR0ORbdGWkPt1vlO2ck089bRe99hfXJ_z9C_TZZhxgTjTSkBZrWc0wyVb_w&h=AT3IMxL9uAtSwd0dcfrJtmZmA5DtlQUDgegZc0T0T0XYxpkILM5CnIVIIKNR_g0PYhQ0udGqbOukfs1OLjvkH6vzzEPwTo2Bi1yZPgA1dnZe1pMGlcZZjpHpv6frbmfia-RCAg)

**AJAX (Asynchronous JavaScript and XML)**:

* AJAX Introduction: [https://www.w3schools.com/xml/ajax\_intro.asp](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.w3schools.com%2Fxml%2Fajax_intro.asp%3Ffbclid%3DIwAR0K7W_0H7Jnzsf32d6QbG-GZG6-bJyL7Yixb9ZhBgW5HKZ2FOhv3QogkuU&h=AT3IMxL9uAtSwd0dcfrJtmZmA5DtlQUDgegZc0T0T0XYxpkILM5CnIVIIKNR_g0PYhQ0udGqbOukfs1OLjvkH6vzzEPwTo2Bi1yZPgA1dnZe1pMGlcZZjpHpv6frbmfia-RCAg)

**User Authentication**:

* PHP User Authentication: [https://www.php.net/manual/en/features.http-auth.php](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.php.net%2Fmanual%2Fen%2Ffeatures.http-auth.php%3Ffbclid%3DIwAR2GEZt7YpKgNZS7ZHXbMjmspFisCx2MpnRLVmjx8Gxrp6oAQLMNF9Eg1IA&h=AT3IMxL9uAtSwd0dcfrJtmZmA5DtlQUDgegZc0T0T0XYxpkILM5CnIVIIKNR_g0PYhQ0udGqbOukfs1OLjvkH6vzzEPwTo2Bi1yZPgA1dnZe1pMGlcZZjpHpv6frbmfia-RCAg)

**Version Control (Git)**:

* Git Documentation: [https://git-scm.com/doc](https://git-scm.com/doc?fbclid=IwAR0K7W_0H7Jnzsf32d6QbG-GZG6-bJyL7Yixb9ZhBgW5HKZ2FOhv3QogkuU)